

AMENDMENTS TO THE CLAIMS

Please amend the claims as indicated in the following listing of all claims:

1. (Currently amended) A method of operating a computer system comprising:
determining if any performance state data stored in the computer system in a first area of
memory specifying performance states for a plurality of processors is associated
with a processor being utilized in the computer system; ~~[[and]]~~
if a portion of the performance state data is found to be associated with the processor
being utilized in the computer system, copying the portion of the performance
state data into a second area of memory; and
if a portion of the performance state data is not found to be associated with the processor
being utilized in the computer system, determining if a near match exists between
a portion of the performance state data and the processor being utilized in the
computer system.
2. (Original) The method as recited in claim 1 where the performance state data is stored
in performance state tables.
3. (Original) The method as recited in claim 1 wherein the method is performed in a
portion of basic input output system (BIOS) and wherein the first area of memory is a memory
segment utilized during power on self test (POST) processing by BIOS and wherein the second
area of memory is a runtime memory segment of BIOS
4. (Original) The method as recited in claim 3, wherein the runtime memory segment
starts at address 0xF000.
5. (Original) The method as recited in claim 1 wherein when the portion of the
performance state data is found to be associated with the processor being utilized in the computer
system, CPUID information associated with the portion of the performance state data matches
CPUID information in the processor.

6. (Canceled)

7. (Currently amended) A method of operating a computer system comprising:
determining if any performance state data stored in the computer system in a first area of
memory specifying performance states for a plurality of processors is associated
with a processor being utilized in the computer system;

if a portion of the performance state data is found to be associated with the processor
being utilized in the computer system, copying the portion of the performance
state data into a second area of memory; and ~~The method as recited in claim 1~~
~~further comprising~~

generating performance state data if none of the performance state data is associated with
the processor being utilized in the computer system and a near match does not
exist.

8. (Original) A method of operating a computer system comprising:
evaluating information in a plurality of performance state tables, each of the performance
state tables having information relating to a performance state of a processor; and
if a match is not found between one of the performance state tables and a processor being
utilized in the computer system, determining if a near match exists and utilizing
one of the performance state tables that is a near match to the processor if the near
match exists.

9. (Original) The method as recited in claim 8, wherein determining if a near match
exists comprises ignoring a stepping field in a CPUID field in the one of the performance state
tables that is the near match to the processor.

10. (Original) The method as recited in claim 8, wherein the plurality of performance
state tables are in a first area of memory, and if the near match is found, copying the information
from the one of the performance state tables into a second area of memory, and wherein the first
area of memory is a memory segment utilized during power on self test (POST) processing by
basic input/output system (BIOS) and wherein the second area of memory is a runtime memory
segment of the BIOS.

11. (Currently Amended) The method as recited in claim 8, further comprising:
if a match is not found between one of the performance state tables and a processor being
utilized in the computer system, and if the near match is determined not to exist,
generating performance state data based on, at least in part, information provided
in the processor.
12. (Original) A method of operating a computer system comprising:
determining if any performance state data stored in the computer system specifying
performance states for a plurality of processors is associated with a processor
being utilized in the computer system; and
generating performance state data if none of the performance state data is associated with
the processor being utilized in the computer system.
13. (Original) The method as recited in claim 12, further comprising generating
performance state data for the processor being used in the computer system after determining
that a near match does not exist.
14. (Original) The method as recited in claim 12, wherein the generating the
performance state data further comprises generating performance state information including a
plurality of frequency values and a fixed voltage value.
15. (Currently amended) The method as recited in claim 14, wherein an upper limit of
frequency of the frequency values is determined according to ~~providing~~ an indication of
maximum frequency provided in the processor.
16. (Original) The method as recited in claim 15, wherein the plurality of frequency
values are spread approximately evenly between a lower limit and the upper limit.
17. (Currently amended) A computer program product stored on computer readable
medium operable in a computer system to:
determine whether a match exists between a processor being utilized in the computer
system and performance state information stored in a first area of memory; [[and]]

if a match exists, ~~copying~~copy matching performance state information into a second area of memory, wherein the first area of memory is a memory segment utilized during power on self test (POST) processing by basic input output system (BIOS) and wherein the second area of memory is a runtime memory segment of BIOS; and if no match is found to exist, generate performance state data after no match is found to exist between any of the performance state data and the processor being utilized in the computer system.

18. (Canceled)

19. (Currently amended) The computer program product as recited in claim 17 [[18]], further comprising generating performance state data for the processor being used in the computer system after determining that a near match does not exist.

20. (Currently amended) The computer program product as recited in claim 17 [[18]], wherein the generating the performance state data further comprises generating a performance state table having a plurality of frequencies and in which processor voltage is fixed.

21. (Currently amended) The computer program product as recited in claim 17 [[18]], wherein an upper limit of the plurality of frequencies is determined in the generated performance state data according to an indication of maximum frequency in the processor and wherein the plurality of frequencies are spread approximately evenly between a lower limit and the upper limit.

22. (Currently amended) A computer program product stored on computer readable medium in a computer system operable to:
determine whether a match exists between a processor being utilized in the computer system and performance state information stored in a first area of memory;
if a match exists, copy matching performance state information into a second area of memory, wherein the first area of memory is a memory segment utilized during power on self test (POST) processing by basic input output system (BIOS) and wherein the second area of memory is a runtime memory segment of BIOS; and

~~The computer program product as recited in claim 17 wherein the computer program product comprises software operable to:~~

determine if a near match exists and utilize performance information from performance state data that constitutes a near match to the processor if the near match exists and if ~~[[a]]~~ the match is not found between any of the performance data and the processor being utilized in the computer system.

23. (Original) A computer program product stored on computer readable medium operable in a computer system to:

determine whether a match exists between a processor being utilized in the computer system and any of a plurality of performance state data associated with respective processors; and

if a match is not found between any of the performance state tables and the processor being utilized in the computer system, the computer program product being operable to determine if a near match exists and utilize performance information from one of the performance state tables that is a near match to the processor if the near match exists.

24. (Original) The computer program product as recited in claim 23, wherein determining if a near match exists comprises ignoring a stepping field in a CPUID field and wherein if the near match exists, copying information from the near match data structure into the second memory area.

25. (Original) The computer program product as recited in claim 23, further comprising code operable to:

generate the performance state data based on information provided in the processor if a match is not found between any of the performance state tables and the processor being utilized in the computer system, and if the near match is determined not to exist.

26. (Original) A computer program product stored on computer readable medium operable in a computer system to:

determine whether a match exists between a processor being utilized in the computer system and performance state information for a plurality of processors; and generate performance state data after no match is found to exist.

27. (Original) The computer program product as recited in claim 26, wherein the generating of the performance state data further comprises generating frequency performance state data in which processor voltage is fixed and frequency changes.

28. (Original) The computer program product as recited in claim 27, wherein an upper limit of frequency is determined in the generated performance state data according to an indication of maximum frequency in the processor.

29. (Original) The computer program product as recited in claim 28, wherein frequency values in the performance state data are spread approximately evenly between a lower limit and the upper limit.

30. (Canceled)

31. (Original) An apparatus comprising:

means for evaluating information in a plurality of performance state tables, each of the performance state tables having information relating to a performance state of a processor; and

if a match is not found between one of the performance state tables and a processor being utilized in the computer system, means for determining if a near match exists and utilizing one of the performance state tables that is a near match to the processor if the near match exists.

32. (Currently Amended) An apparatus comprising:

means for determining if [[any]] performance state data stored in the computer system specifying performance states for a plurality of processors is associated with a processor being utilized in the computer system; and

means for generating performance state data if none of the performance state data is associated with the processor being utilized in the computer system.

33 - 42. (Canceled)